Open-Source Development with Intel AI Retail Suite Project

Notices and Disclaimers

- Intel technologies may require enabled hardware, software or service activation.
- No product or component can be absolutely secure. Your costs and results may vary. Results have been estimated or simulated.
- Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex
- Intel is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's Global Human Rights Principles. Intel's products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.
- Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.
-

Agenda

- Introduction
- CV-based AI/ML Retail Use Cases
- AI Retail Suite Project
- Workshop
- Project GitHub + Compliances
- Project Contributions
- Conclusion
- Take-Home Hackathon – Deadline Monday 27th 11:59 pm - Midnight

Team/Speakers

Neethu Elizabeth Simon
Senior Software Engineer
Intel Corporation

Antonio Martinez
Senior Software Engineer
Intel Corporation

Brian McGinn
Senior Software Engineer
Intel Corporation

Elizabeth Lee
Senior Software Engineer
Intel Corporation

Introduction

- New and innovative markets
- Increasing industry support
- Strong drivers:
  - Cheaper and faster processors and wireless networks
  - AI/ML Advancements

INTERNET OF THINGS IS A GROWING WEB OF NUMEROUS DEVICES THAT ARE INTERCONNECTED AND INTERACTING

NUMBER OF "THINGS" GETTING INSTALLED AND CONNECTED IS GROWING

Global IoT Market worth – $153.2B by 2029, growing at 18.8%

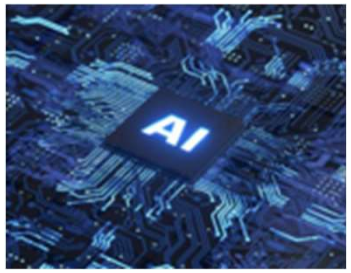
https://www.marketsandmarkets.com/Market-Reports/internet-of-things-market-573.html

CV-based AI/ML Retail Use Cases

- Camera is the ultimate "Thing"
- CV + AI = "Eye of IOT"
- Smart Retail Use Cases

AI Challenges

What are Some of the AI Challenges Today?



GPU Availability

AI is Heterogenous and alternatives exist

Vendor Lock-in

Avoid vendor lock-in with open-source standards-based software

Cost

Need better price and performance across hardware and software portfolio

Secure AI

Set the trust boundary appropriate to your workloads

Intel Retail AI Suites - Automated Self-Checkout Project Performance Benchmarcking Retail Use Cases

Our Objective

Make it Easy to Work with AI on Intel®

Hardware

- Most cost optimized hardware
- Recommend hardware for scale
- Accurate ROI calculations

Software

- Software for bootstrapping
- Customize AI pipelines per use case
- Benchmarking custom workloads

Build an Engaged Partner Community

GitHub

https://github.com/intel-retail

Discord

https://discord.gg/2SpNRF4SCn

Automated Self-Checkout Reference Implementation

Helping Build Vision Enabled Systems

Automated Self-Checkout Retail Reference Implementation

- Source Code
- Microservices
- Documentation
- Learning Videos

Benchmark Script | Hardware Recommandation | Pre-Trained Models

Tools & Libraries

OpenVINO™

Intel® OpenVINO Model Server to run inferencing across many frameworks with optimised performance

docker

Operating System Support

ubuntu

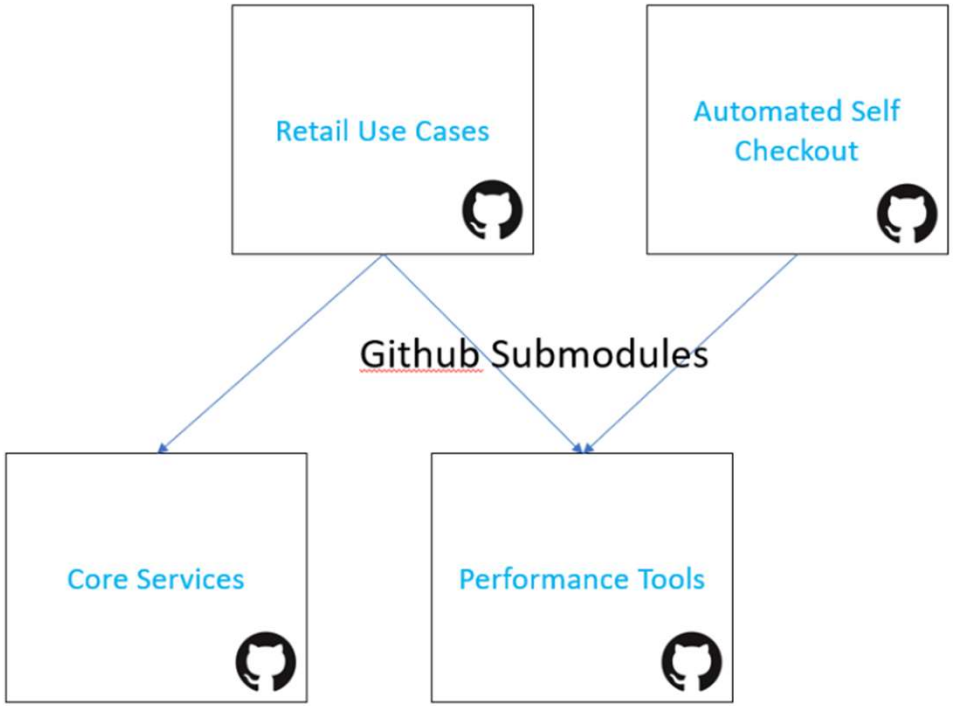
Support for Platforms Based on Intel® Architecture

intel. CORE i7 | intel. CORE i9 | intel. DATA CENTER GPU FLEX SERIES | intel. ARC GRAPHICS | intel. XEON

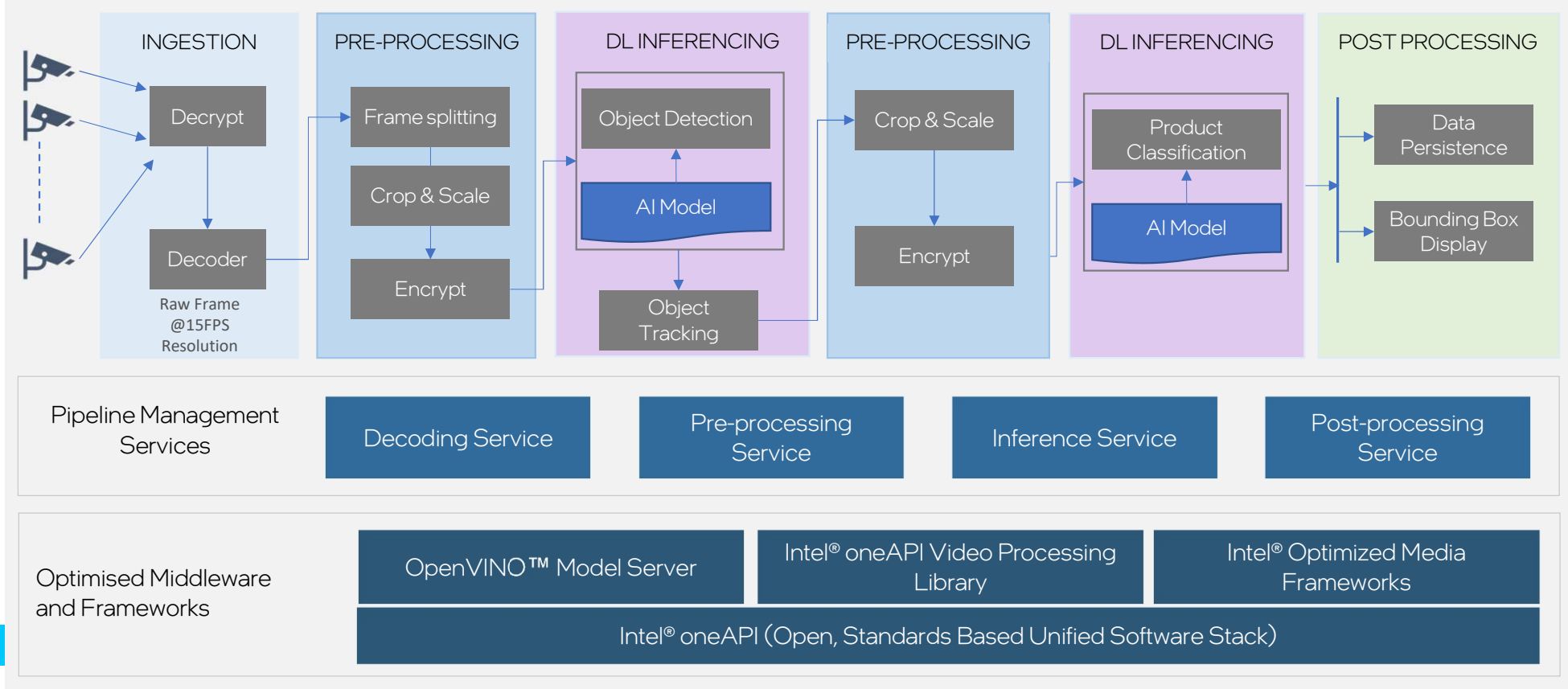
intel.

Automated Self-Checkout Reference Implementation

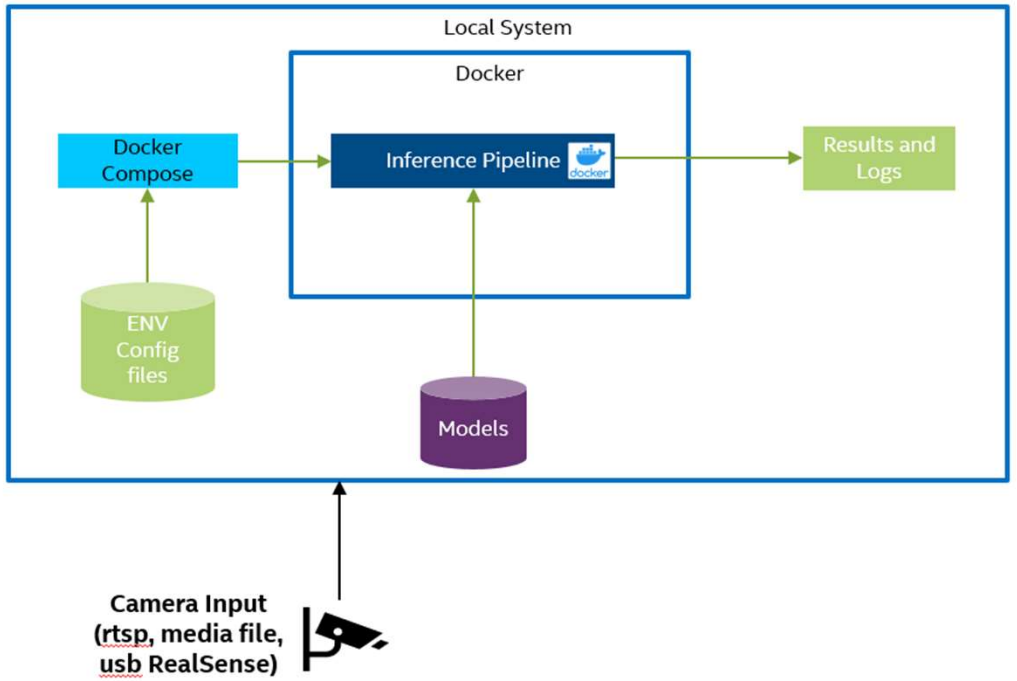
https://github.com/intel-retail/



Vision Inference Pipeline (Object detection + Classification)



Architecture



Legend

- Docker Containers
- Docker yml
- Text File
- AI Model
- File System Operation
- gRPC Request
- USB/Network Request

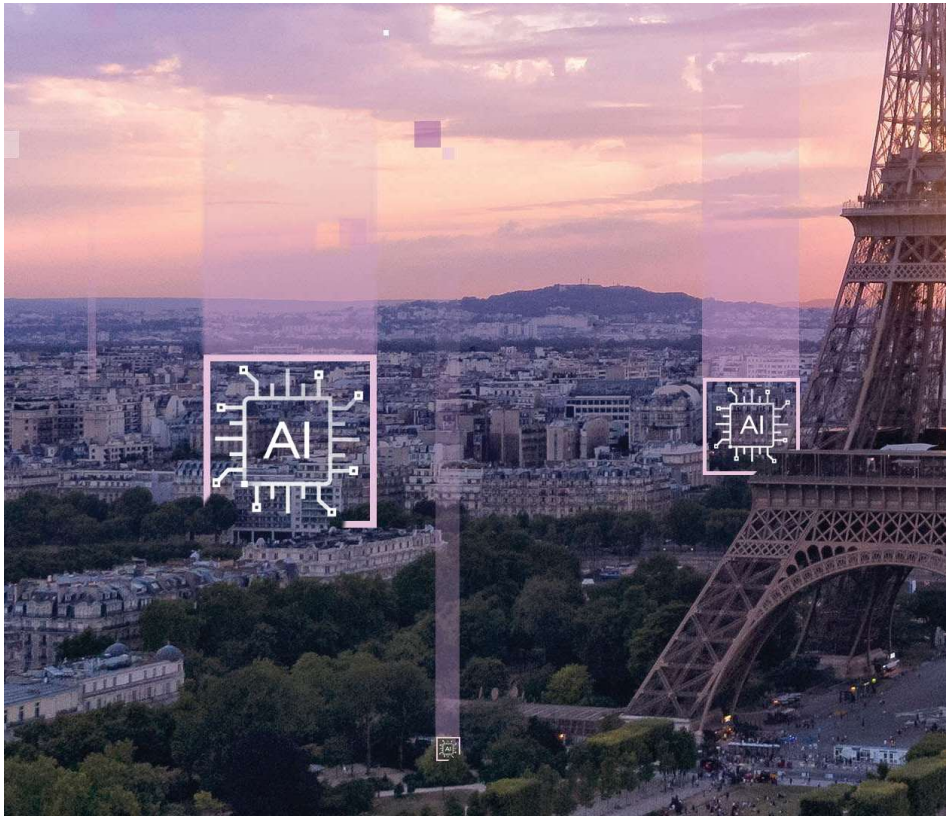
Workshop

Pre-requisites

- OS – Preferably Ubuntu 22.0
  - Can work on lower versions, however latest features are not tested there
  - If using Windows, then can use Ubuntu on Windows via WSL
    https://learn.microsoft.com/en-us/windows/wsl/install
- Hardware – Intel CPUs (minimum core 8th gen, can have Intel GPUs too)
- Software –
  - Docker (Tested on version >= 23.0.0)
  - Docker Compose v2
  - Git
  - Make - install with apt install make
  - OpenVINO installation – https://github.com/openvinotoolkit/openvino_notebooks?tab=readme-ov-file#-installation-guide
- Optional – Camera (in-built or separate) for live feed

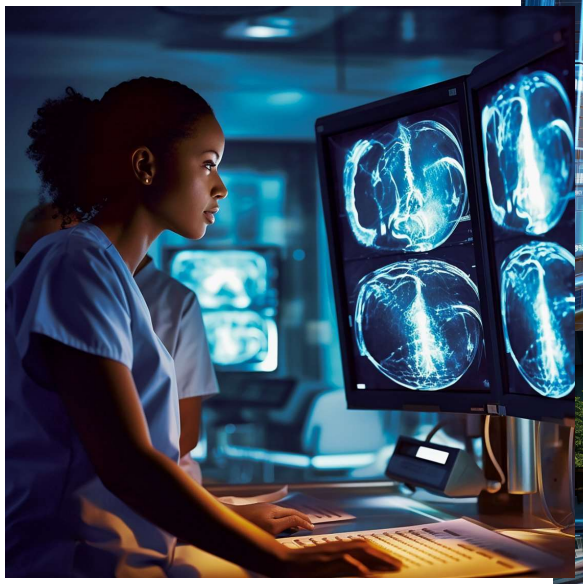
Definitions

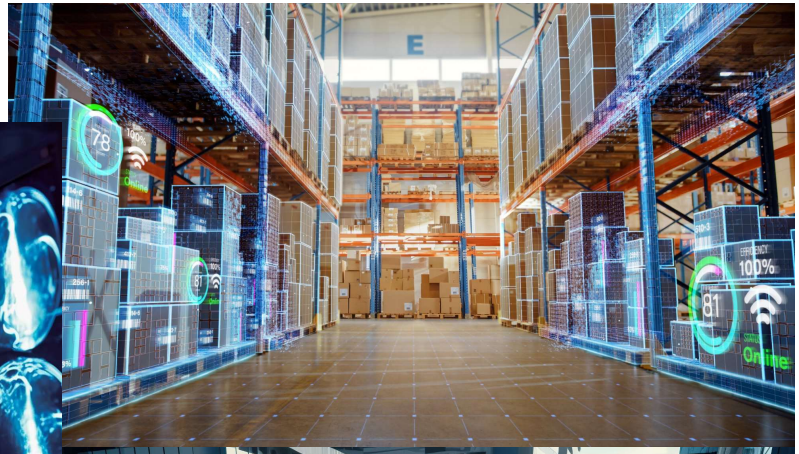
- AI involves teaching computers to process data in a way that mimics the human brain
- Computer Vision is a field of AI that focuses on enabling computers to interpret visual data
- AI and CV are important in our lives for several reasons:
  - Automation
  - Medicine
  - Safety
  - Customer experience

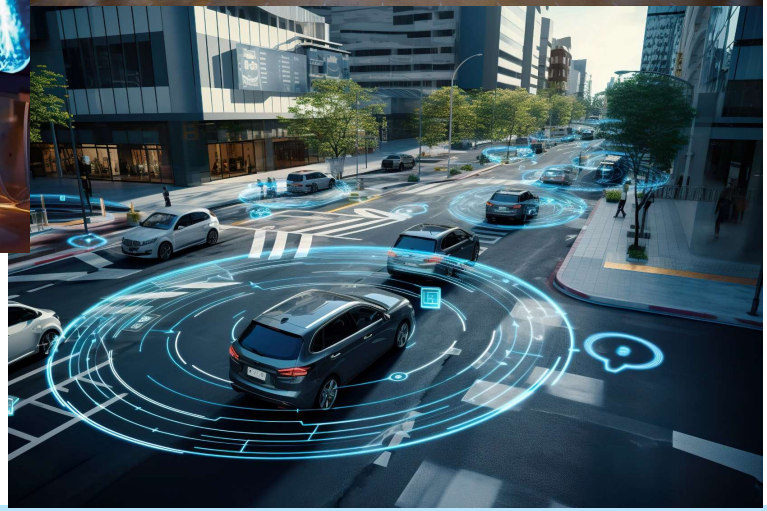


Computer Vision Use cases

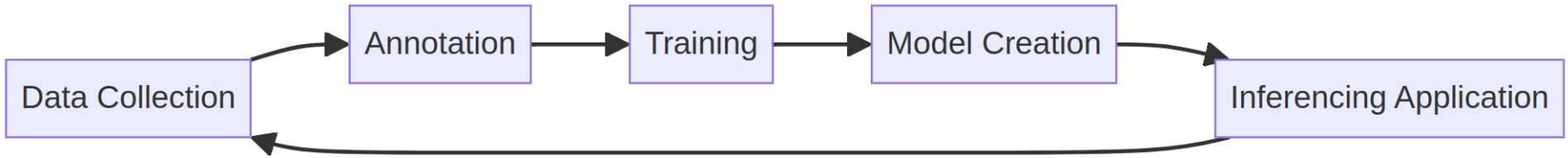
- Robotics
- Retail
- Surveillance and Security
- Medical Imaging

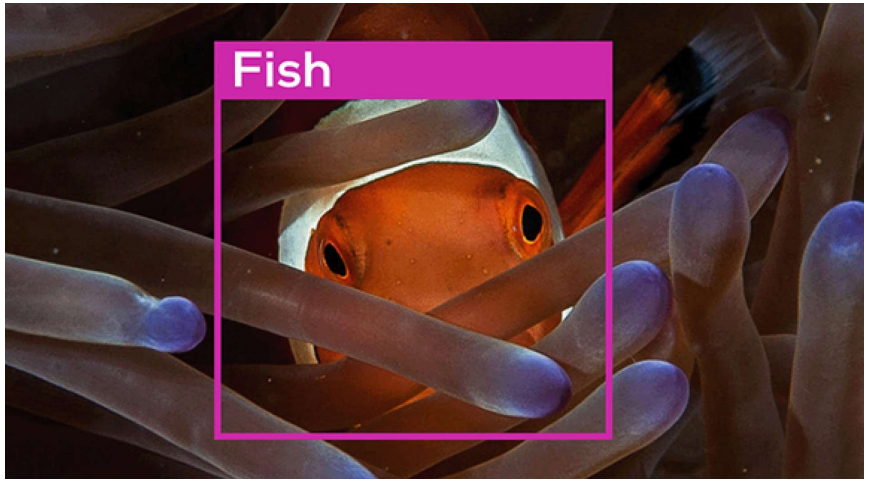






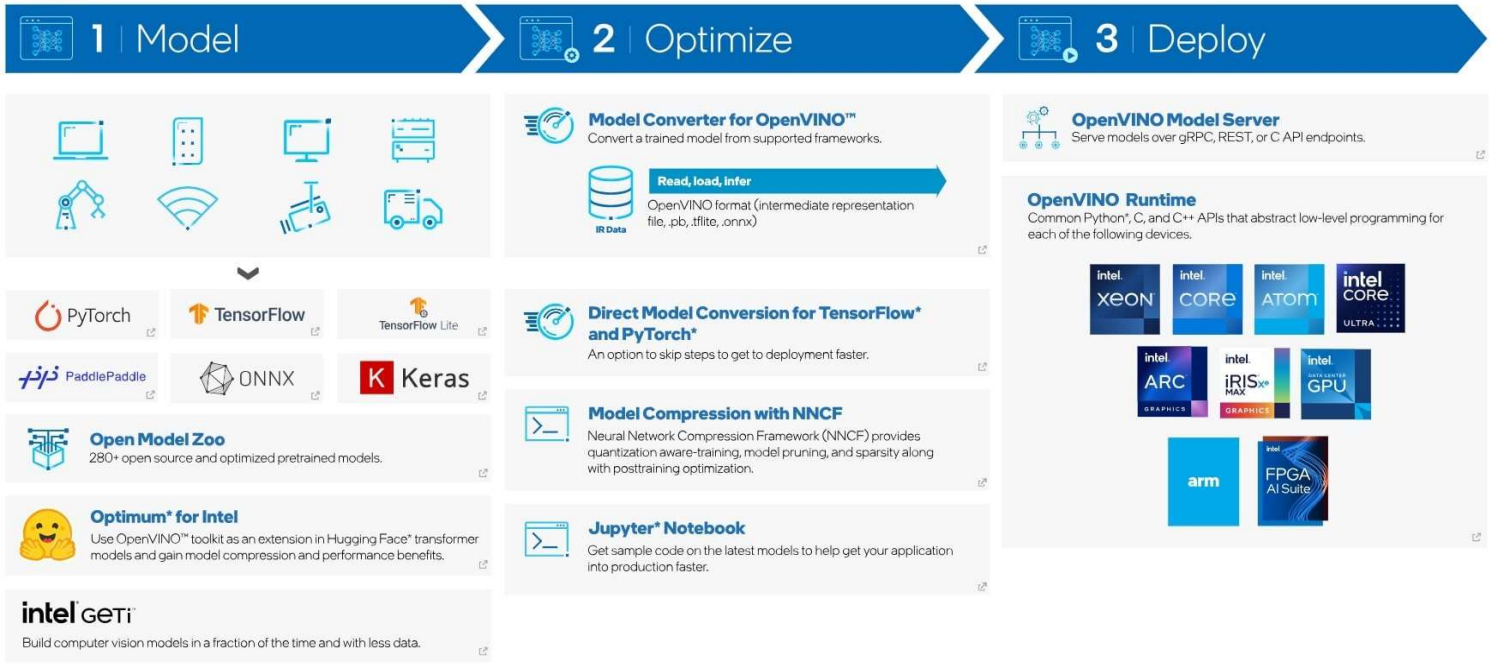
Pre-train AI Models





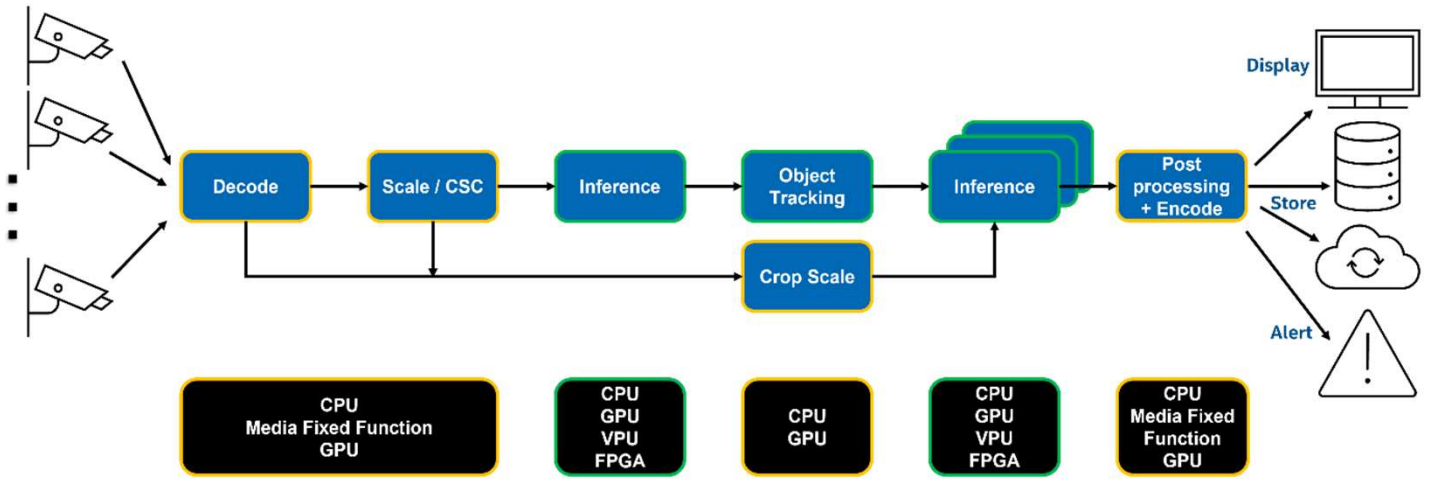
Intel ® Distribution of OpenVINO ™ Toolkit

- Opensource toolkit that accelerates AI inference.
- Used for AI development and integration of deep learning in domains like computer vision, LLM and GenAI.

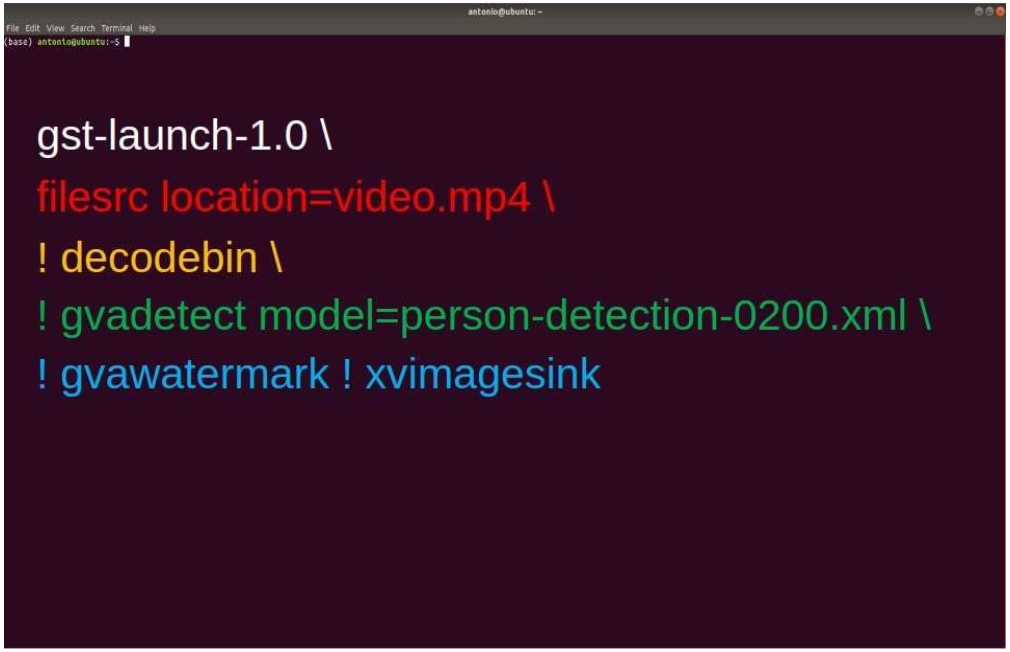


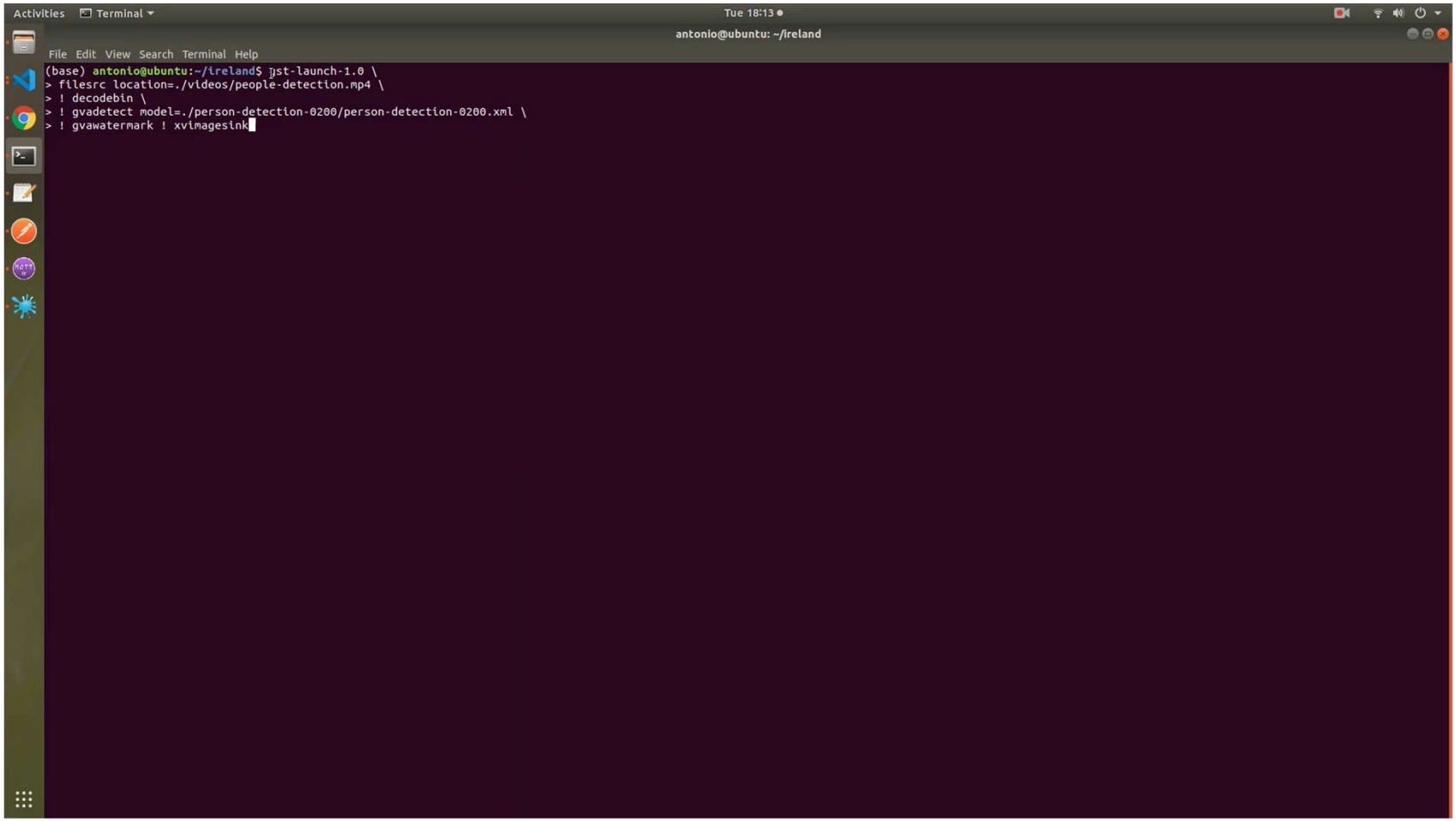
Intel ® DLStreamer

- Streaming media analytics framework, based on Gstreamer + OpenVINO inference engine.
- Enables developers to create deep learning pipelines across Intel architecture.

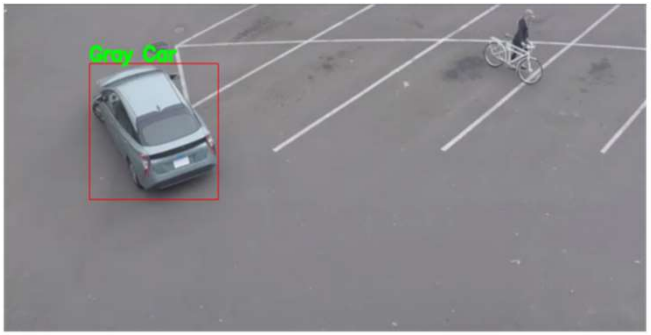


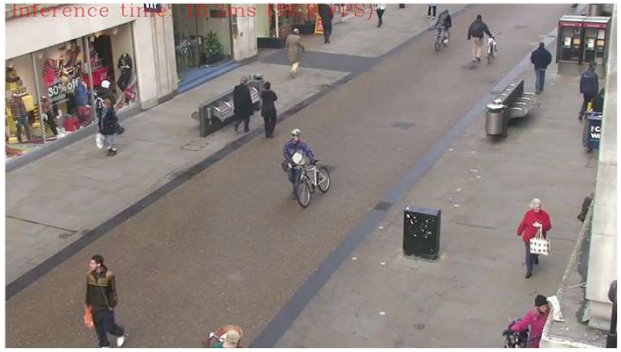
Intel ® DLStreamer Demo

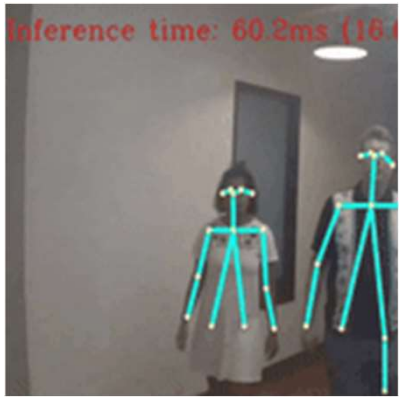



Activities
Terminal
Tue 18:13
antonio@ubuntu: ~/ireland
File Edit View Search Terminal Help
(base) antonio@ubuntu:~/ireland$ gst-launch-1.0 \
> filesrc location=./videos/people-detection.mp4 \
> ! decodebin \
> ! gvadetect model=./person-detection-0200/person-detection-0200.xml \
> ! gvawatermark ! xvimagesink

OpenVINO Jupyter Notebooks







GenAI

- Text to Video/Image generator
- Large Language Models (LLM)
- Visual-Language Models (VLM)
- Sound generators

Darth Vader surfing on waves

A beautiful pink unicorn

Performance Benchmarking

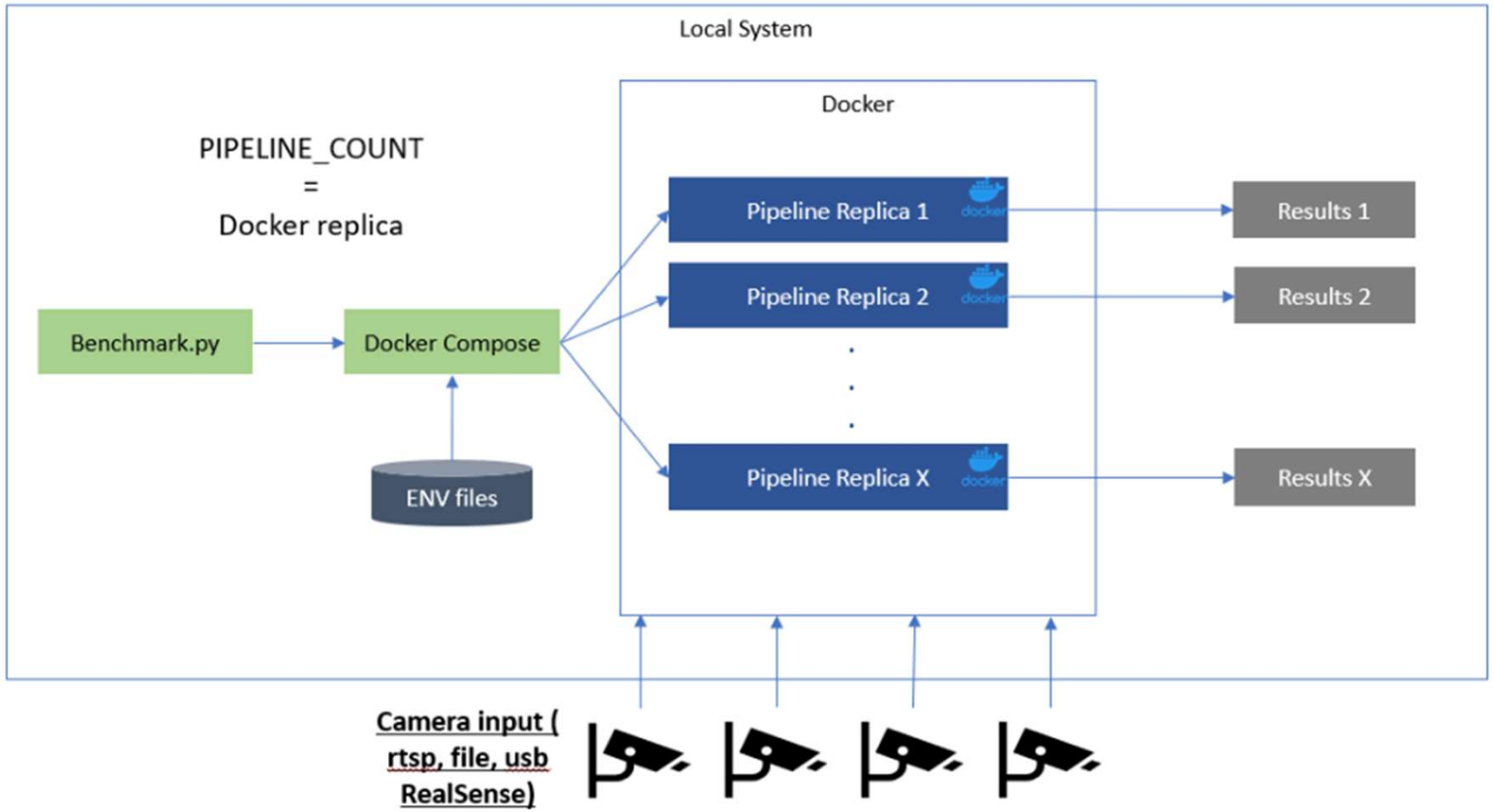
Performance Benchmarking

- You can benchmark pipelines with a collection of scripts to get the pipeline performance metrics such as video processing in frame-per-second (FPS), memory usage, power consumption, and so on.

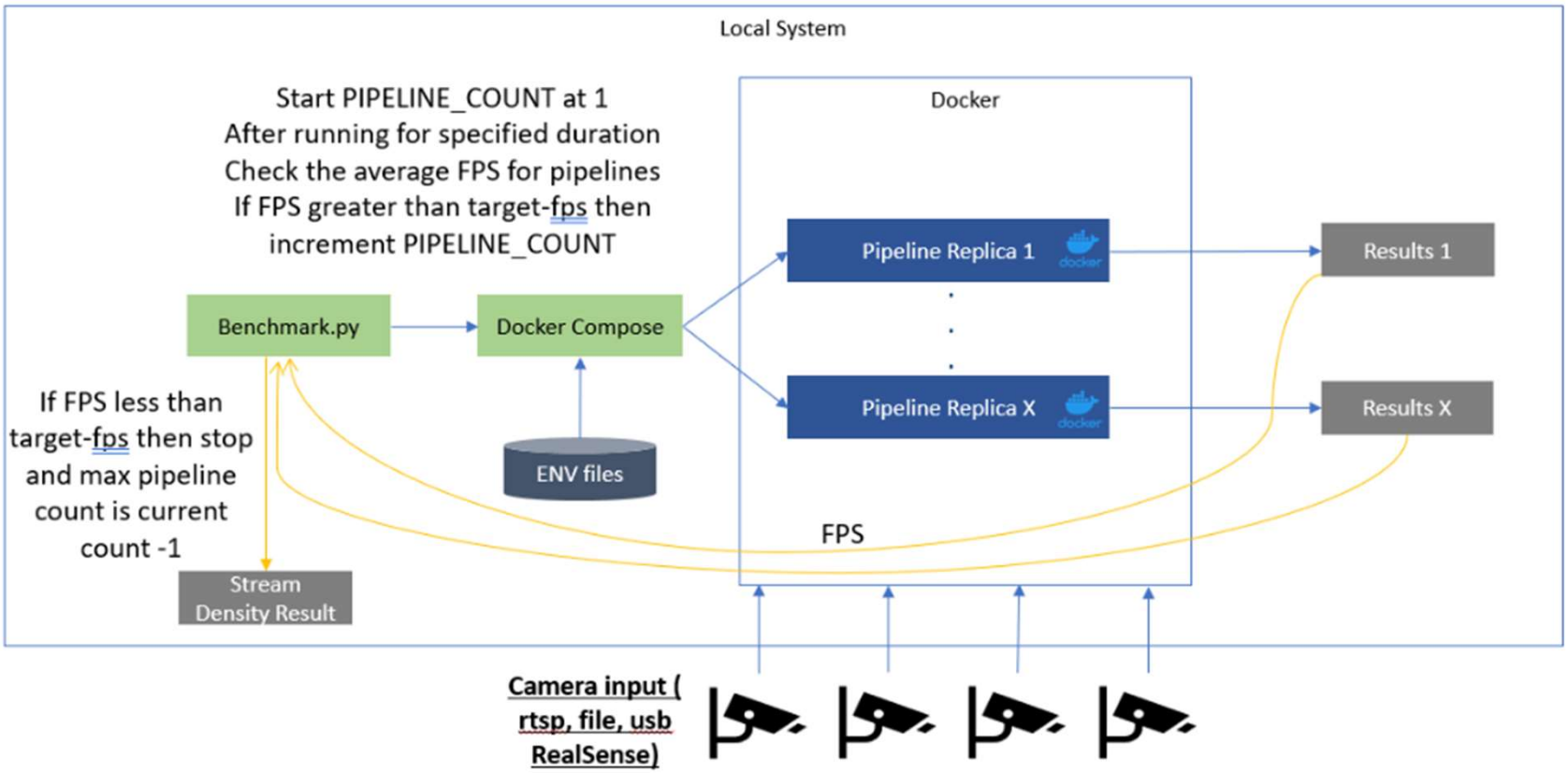
```
Stream density TARGET_FPS set for 14.9 and INIT_DURATION set for 30
Starting single container stream density benchmarking
Starting pipeline: 1
waiting for pipelines to settle
FPS for pipeline0: 30.9545
Total FPS throughput: 30.9545
Total FPS per stream: 30.9545
Starting pipeline: 3
waiting for pipelines to settle
FPS for pipeline0: 14.668
FPS for pipeline1: 14.0375
FPS for pipeline2: 14.0805
Total FPS throughput: 42.786
Total FPS per stream: 14.262
Starting pipeline: 2
waiting for pipelines to settle
FPS for pipeline0: 25.384
FPS for pipeline1: 36.1835
Total FPS throughput: 61.5675
Total FPS per stream: 30.7838
Max stream density achieved for target FPS 14.9 is 2
Finished stream density benchmarking

,Metric,10objs
0,Total Text count,NA
1,Total Barcode count,NA
2,Camera_0 FPS,15.786
3,Camera_0 Last log update,08/24/2023 00:04:858186
4,CPU Utilization %,11.3
5,Memory Utilization %,28.593
6,Disk Read MB/s,0.001
7,Disk Write MB/s,0.002
8,S0 Memory Bandwidth Usage MB/s,3487.143
9,S0 Power Draw W,14.227
10,GPU_0 VDBOX0 Utilization %,5.736
11,GPU_0 GPU Utilization %,38.864
```

Performance Architecture



Stream Density



Bring Your Own AI Pipeline and Integrate with Intel Retail AI Suite for Performance Benchmarking

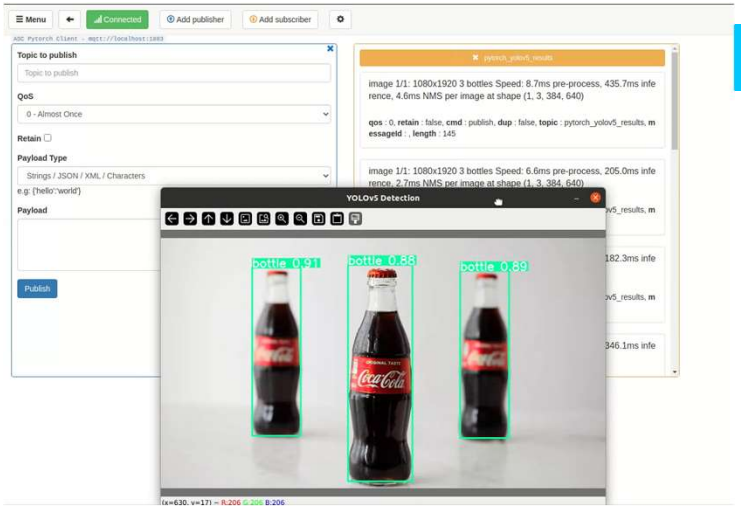
Integration Steps

- Build your own AI/ML pipeline
- *git clone https://github.com/intel-retail/retail-use-cases.git*
- Add Your AI code to intel-retail/retail-use-cases repo – https://github.com/intel-retail/retail-use-cases/tree/main/use-cases/demos

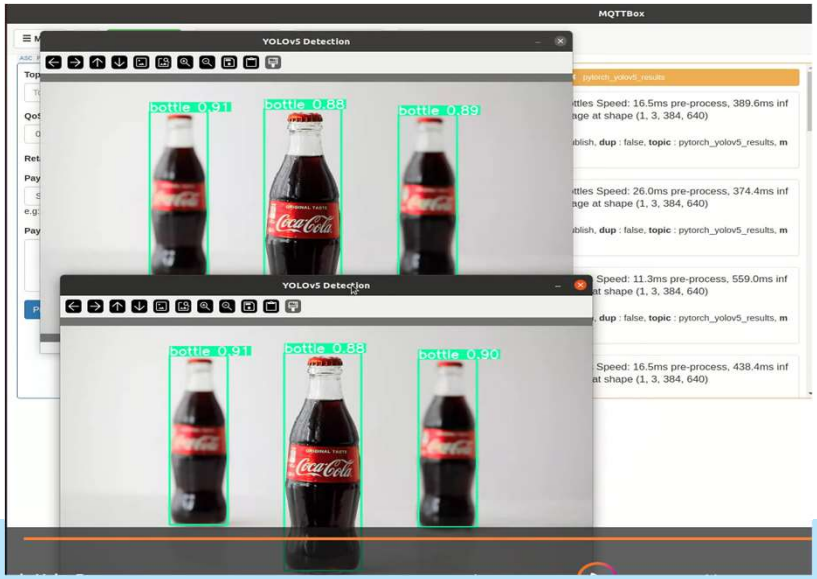
Integration Steps

- *Build pipeline – make build_pytorch*
- *Run 1 pipeline – make run_pytorch_object_detection*
- *Run 2 pipeline –*

PIPELINE_COUNT=2 make run_pytorch_object_detection



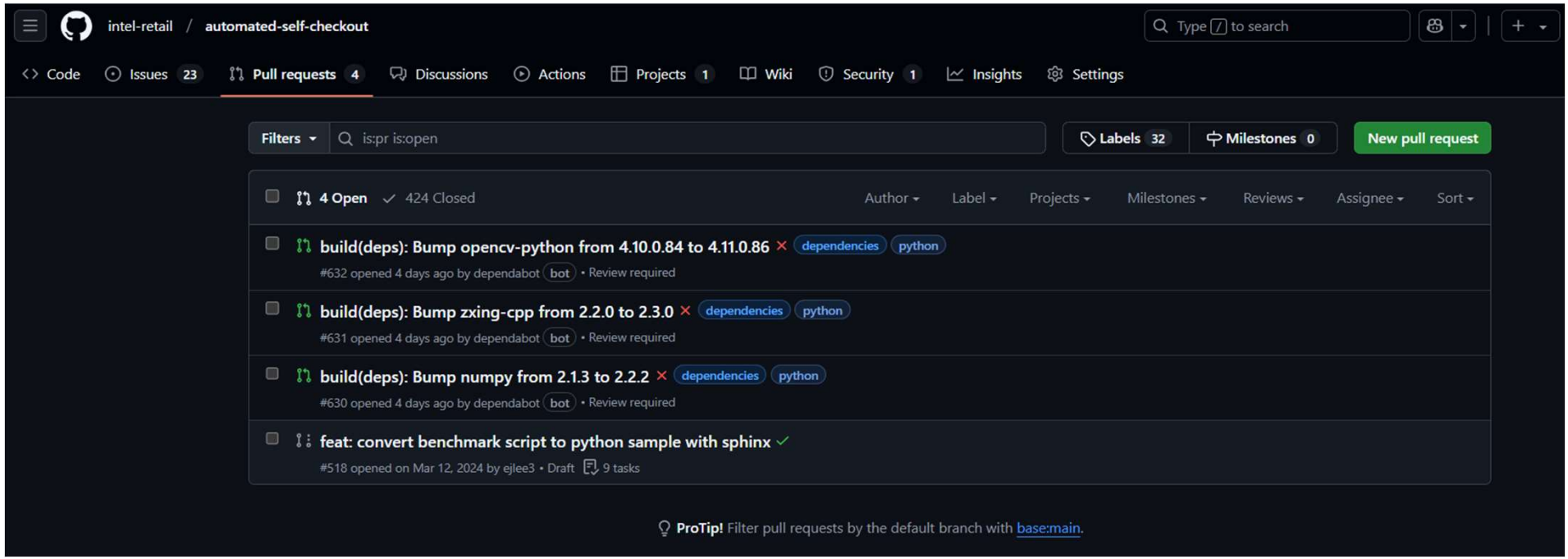
- Start MQTT client to view results – use MQTTBox, MQTTX or cmd
- Result and Performance logs under results folder



GitHub: GitHub Actions, Security Review Tools, and the PR Code Review Process - Brian

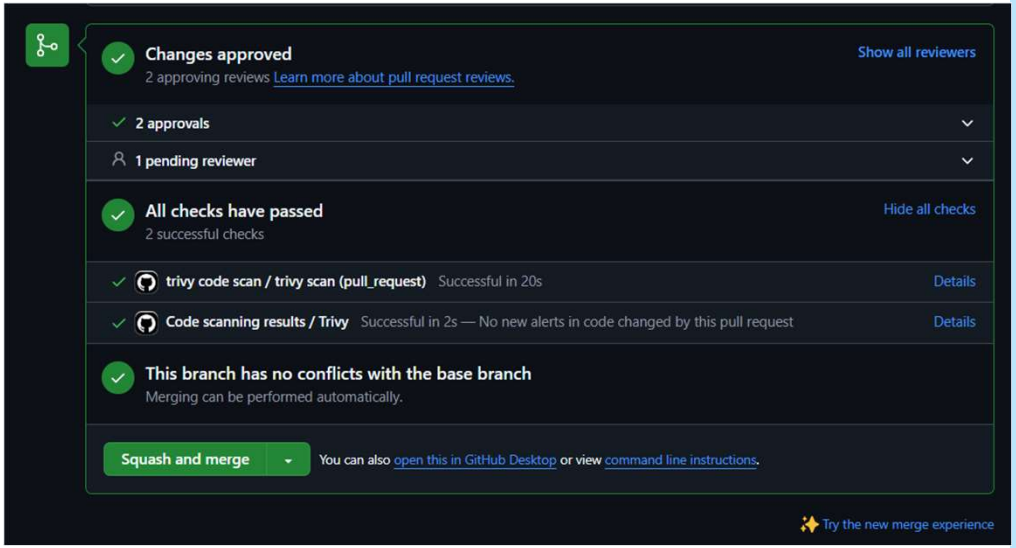
Pull Request

https://github.com/intel-retail/automated-self-checkout/pulls



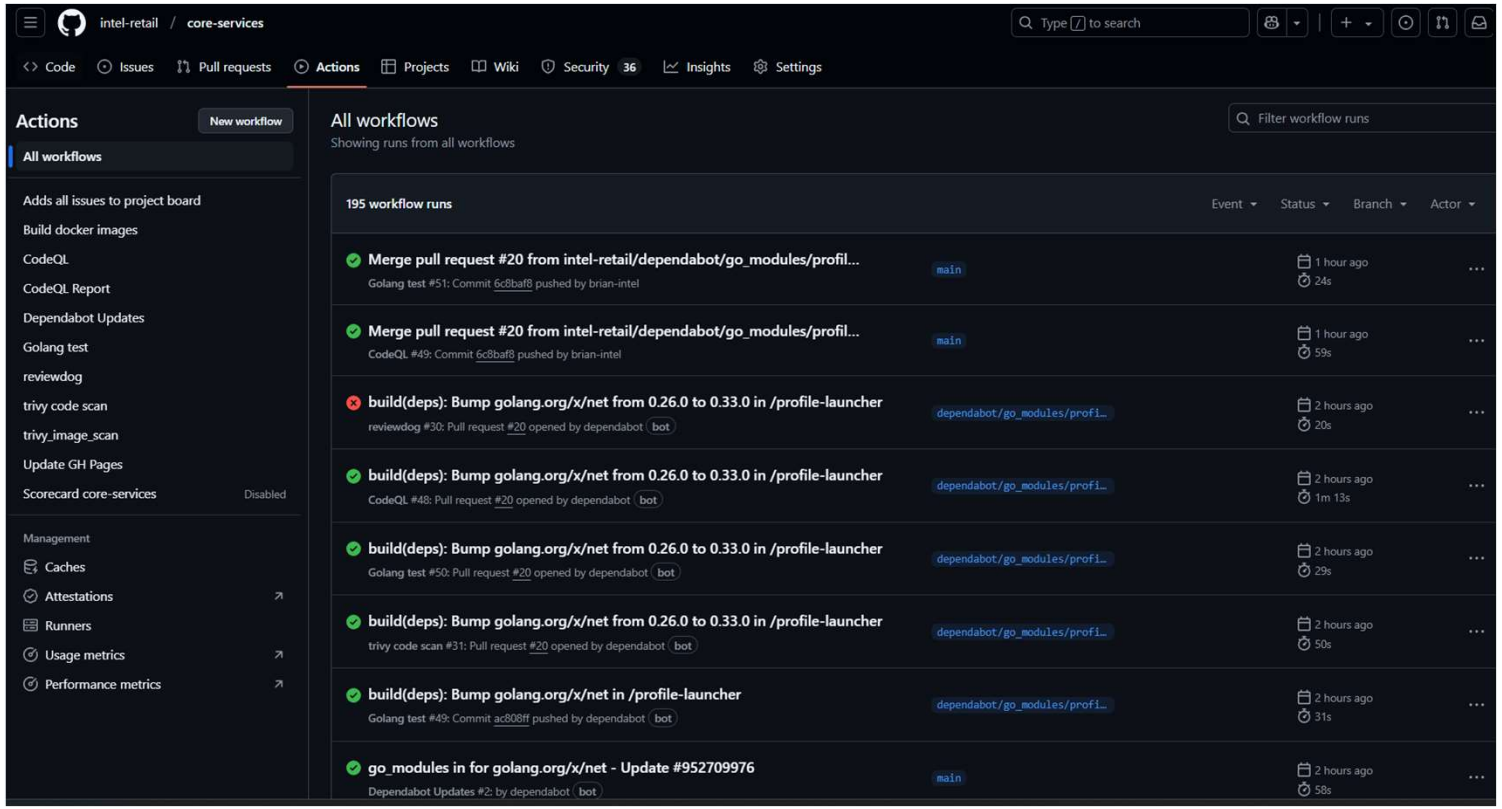
Pull Request Validation

- Requires 1 or more review approvals
- Github Actions automatically validate all pull requests
- Validation must pass before merging



Github Actions

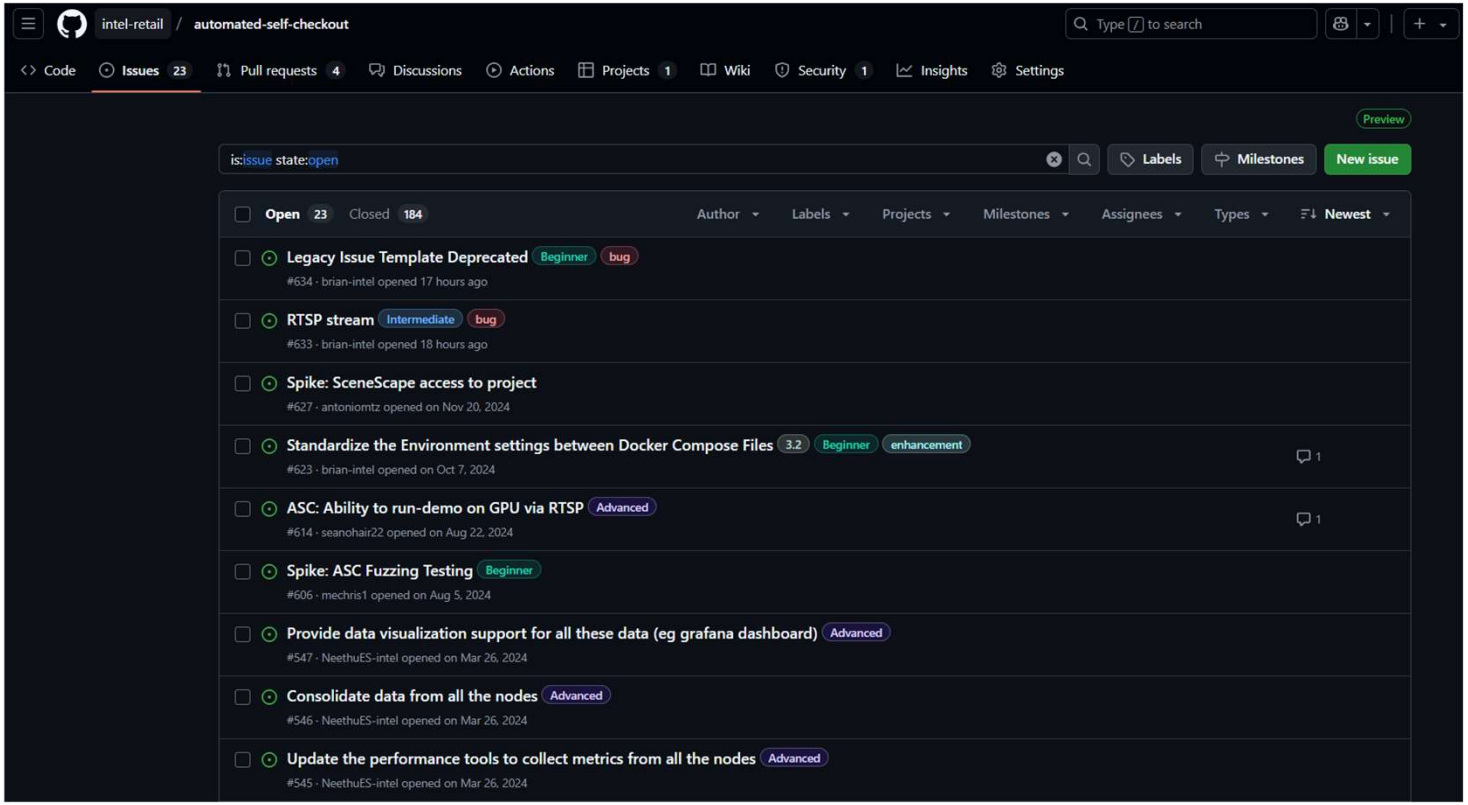
https://github.com/intel-retail/core-services/actions



Overview of the Project Issue/Feature List, how to make Requirement/Feature Requests, and Picking Up Issues - Brian

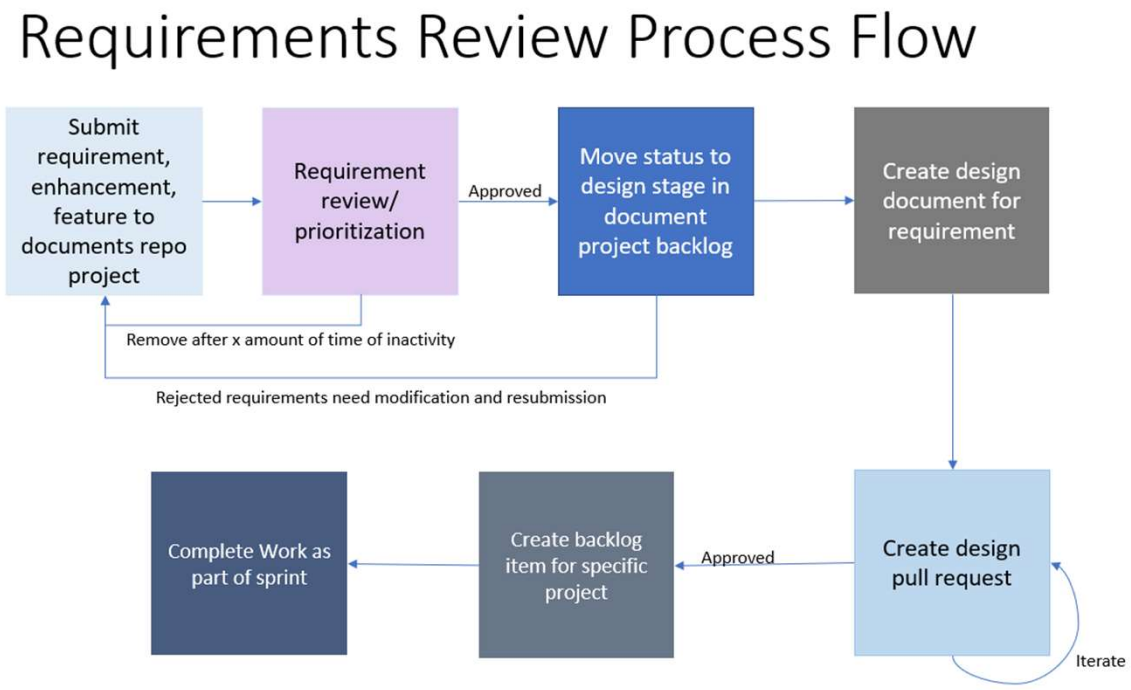
Open Issues

https://github.com/intel-retail/automated-self-checkout/issues



Submitting a requirement

https://github.com/intel-retail/documentation



Future Features

- Add multi-camera item tracking to the loss prevention pipeline
- Add additional configurations for the latest Intel hardware (ex. NPU support)
- Continual learning and retraining to complement our pre-trained models

Conclusion

Community Call to Action

- Fork the repository
- Look into contributing
- File issues and requirements
- Join our Discord for communication

GitHub

https://github.com/intel-retail

Discord

https://discord.gg/2SpNRF4SCn

Take-Home Coding Hackathon - Rules

- Team participation - can have between 2 to 3 members in your team.
- Take-Home Hackathon starts now
- **Submission Deadline – Monday 27th, Midnight 11:59 pm**
- Prizes to be won – 1st ,2nd ,3rd and Rookie Award (Undergrads)

GitHub

https://github.com/intel-retail

Discord

https://discord.gg/2SpNRF4SCn

Take-Home Coding Hackathon – What/How

- Select any repo under intel-retail
- Select any issues to work on from the list under Issues tab or create your own issue as New issue to work on
- Existing issues labeled as – Beginner, Intermediate, Advanced
- Make sure to assign the issue to your group
- Fork the repository to your account
- Work on the issue
- Create PR to upstream your changes
- Submit as many PRs as you like within the Deadline
- For any questions, either comment on the issues or reach out in Discord

Take-Home Coding Hackathon – Judging Rules

- The judges will evaluate your PRs on GitHub.
- Expect some back-and-forth communication to happen to enable the upstreaming of your PR.
- When your PR is merged, the judges will score the PR based on the quality of your contributions.
- They will use the "Judging Criteria" to score the contributions.
- The judge's scores are final.
- Results will be announced by Monday, February 3rd

Take-Home Coding Hackathon – Judging Criteria

- Feel free to contribute to any of the categories below.
- Based on the category of your submission, the Judges will score your contributions based on the **complexity, completeness, and code/documentation quality**.
- The judges' scores are final.
- The maximum score per category is listed below:

Criteria	Max Score
Bug Fix	40
Documentation	20
Feature Implementation (Business Value(20)/ Architecture(20)/ Innovation(20)/ Code Quality(20)/ User Experience(20))	100
Unit Tests	30
Integration Tests	30
CI CD/Compliance/Automation	30

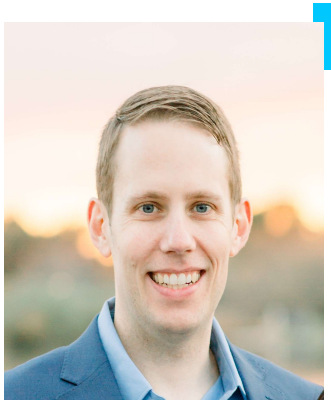
Thank You

Questions?

Join & Build with Us

Neethu Elizabeth Simon
Senior Software Engineer
Intel Corporation

Antonio Martinez
Senior Software Engineer
Intel Corporation



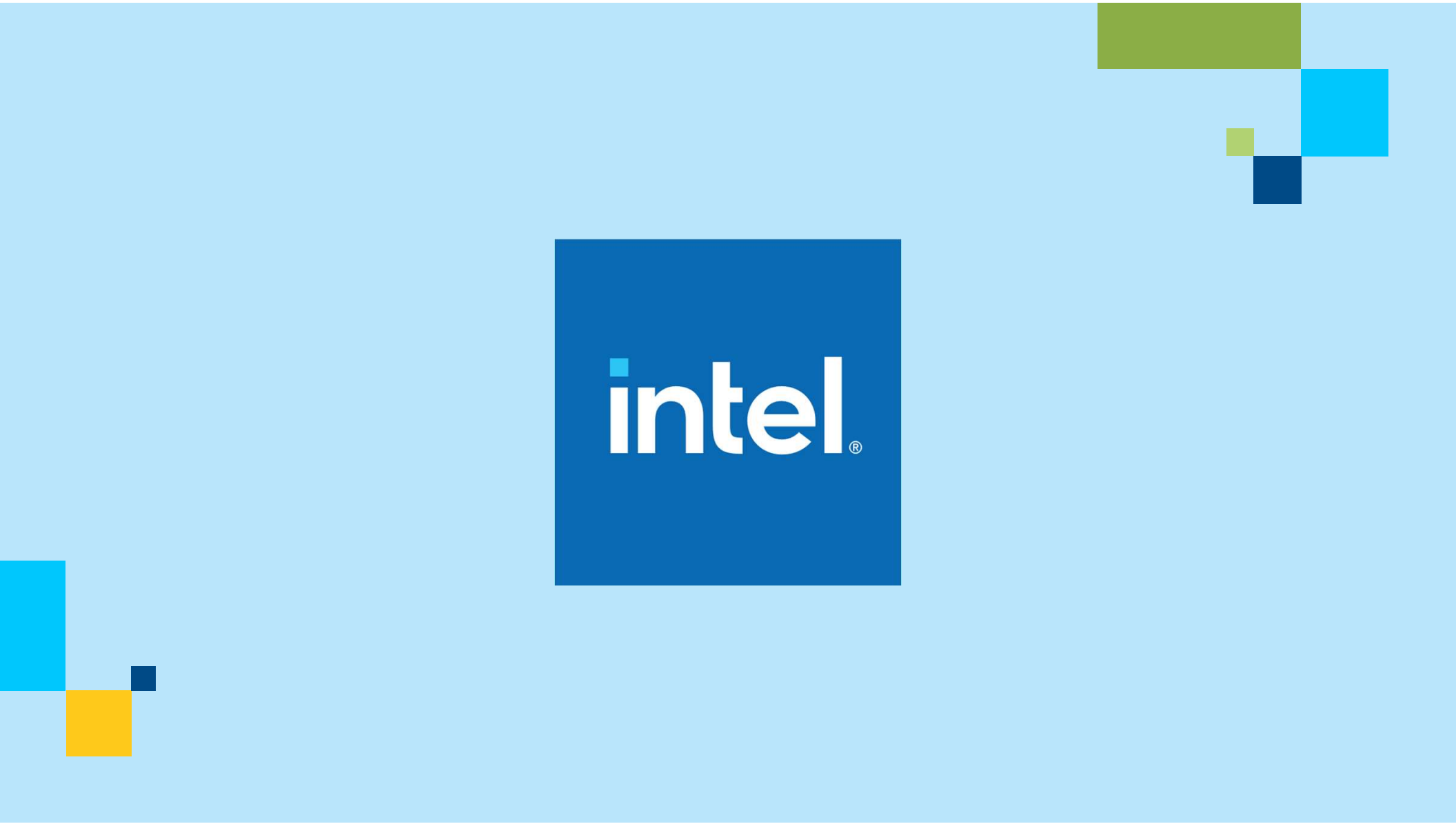
Brian McGinn
Senior Software Engineer
Intel Corporation

GitHub

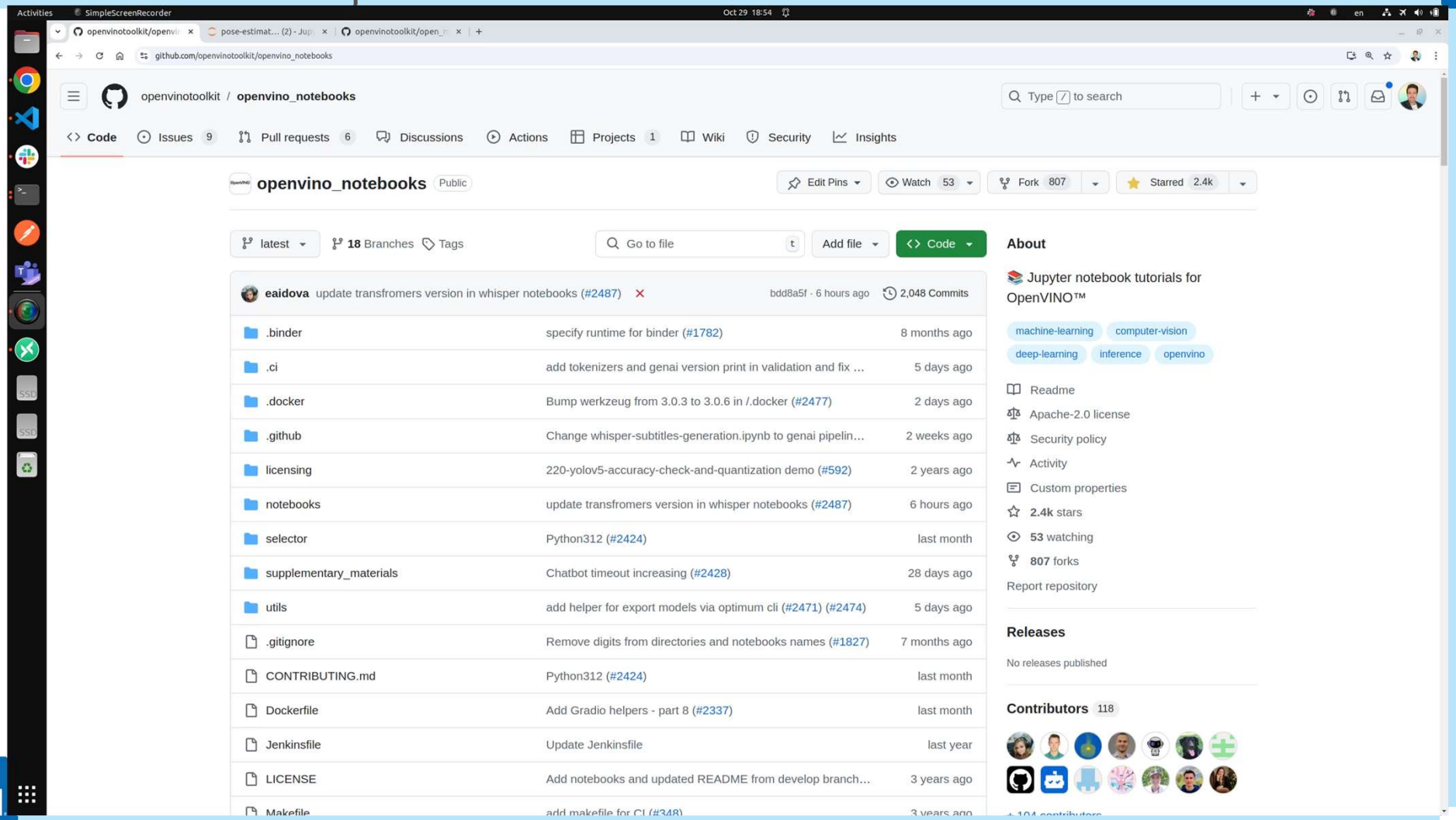
https://github.com/intel-retail

Discord

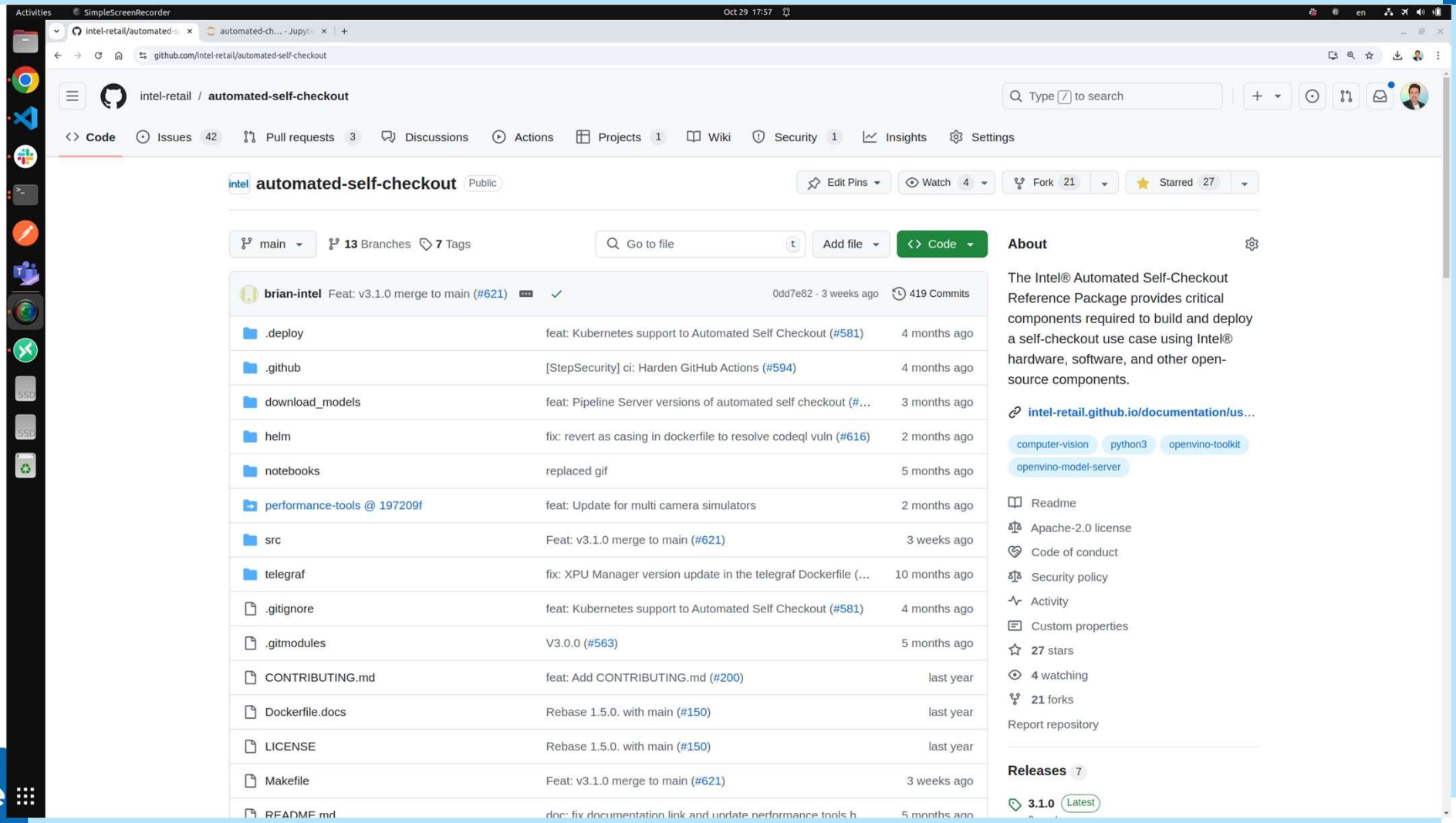
https://discord.gg/2SpNRF4SCn

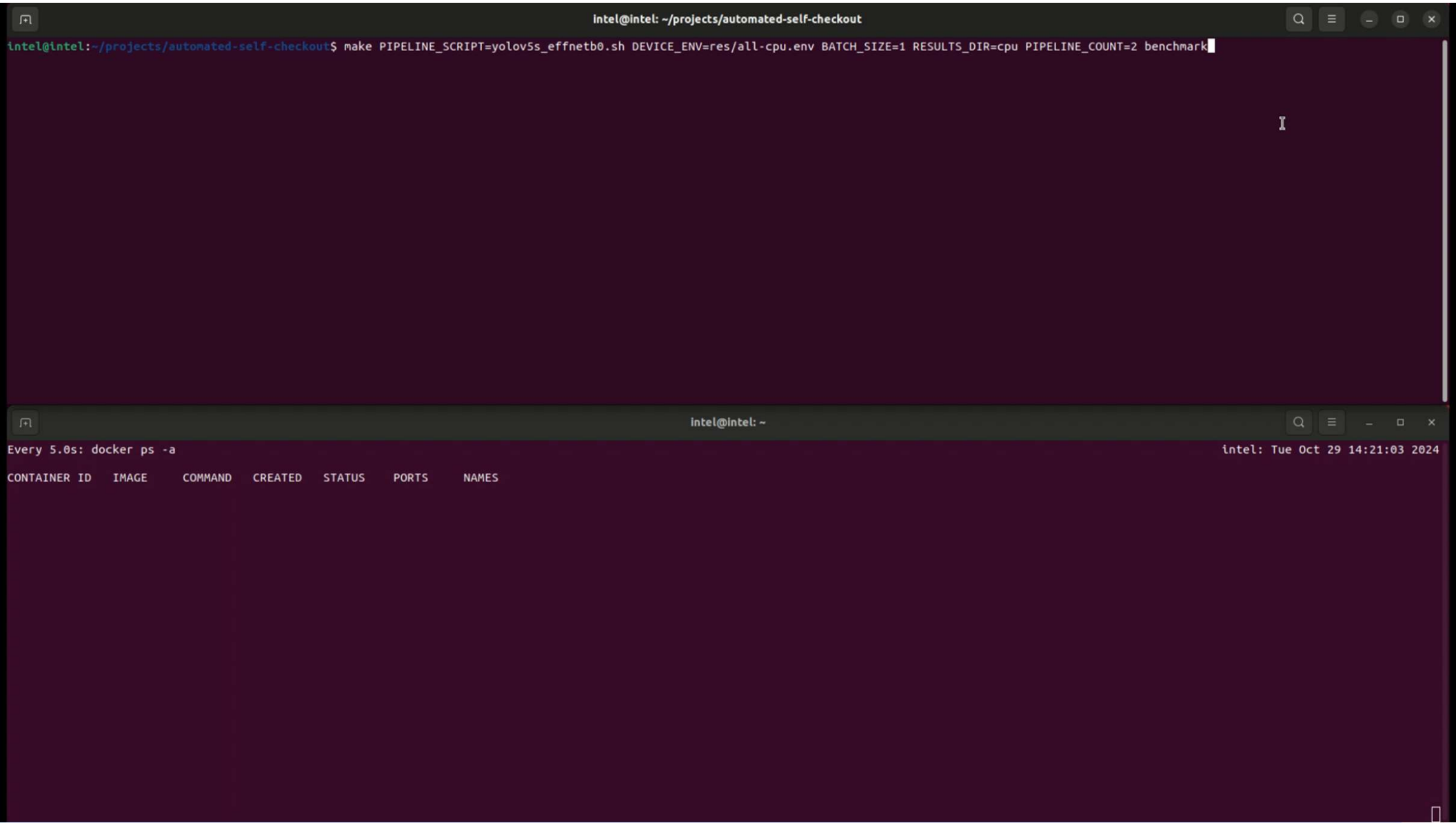

intel

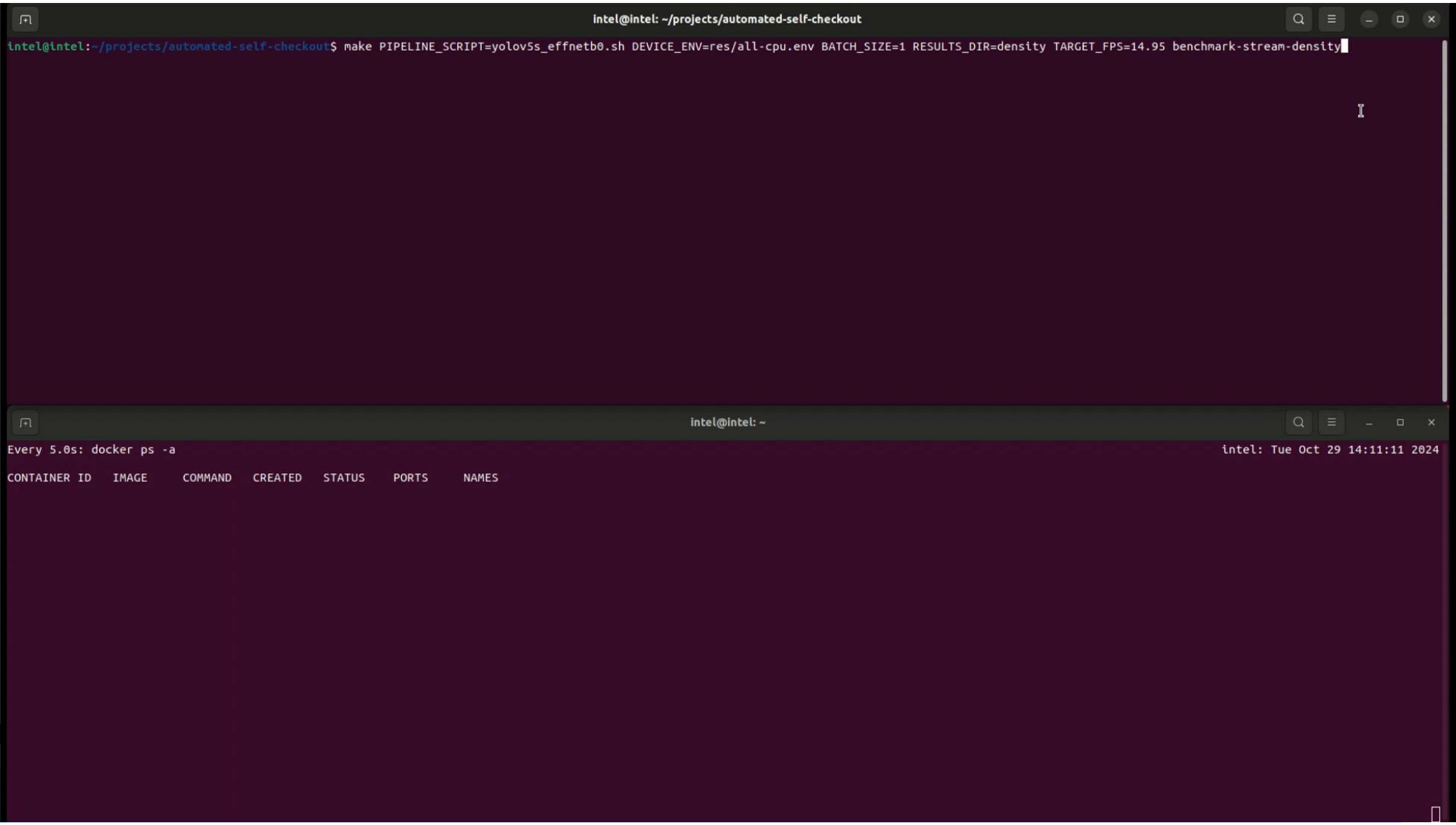
OpenVINO Notebooks



Automated Self-Checkout




intel@intel: ~/projects/automated-self-checkout
intel@intel:~/projects/automated-self-checkout$ make PIPELINE_SCRIPT=yolov5s_effnetb0.sh DEVICE_ENV=res/all-cpu.env BATCH_SIZE=1 RESULTS_DIR=cpu PIPELINE_COUNT=2 benchmark
intel@intel: ~
Every 5.0s: docker ps -a
intel: Tue Oct 29 14:21:03 2024
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES


intel@intel: ~/projects/automated-self-checkout
intel@intel:~/projects/automated-self-checkout$ make PIPELINE_SCRIPT=yolov5s_effnetb0.sh DEVICE_ENV=res/all-cpu.env BATCH_SIZE=1 RESULTS_DIR=density TARGET_FPS=14.95 benchmark-stream-density
intel@intel: ~
Every 5.0s: docker ps -a
intel: Tue Oct 29 14:11:11 2024
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES